UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/079,811	02/22/2002	Andrew Mark Nightingale	550-318	5550

23117 7590 11/29/2006

NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
PATEL, SHAMBHAVI K	

ART UNIT	PAPER NUMBER
2128	

DATE MAILED: 11/29/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/079,811
Filing Date: February 22, 2002
Appellant(s): NIGHTINGALE ET AL.

Stanley C. Spooner
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 20 September 2006 appealing from the Office Action mailed 21 March 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,182,258	**Hollander**	01-2001
5,835,764	**Platt**	11-1998
6,810,373	**Harmon**	10-2004
6,408,009	**Campbell**	06/2002

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

1. **Claims 1, 2, 6, 7-12, 14, 15, and 16** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Hollander (US Patent No 6,182,258)** in view of **Platt et al (US Patent No. 5,835,764)**, herein "Platt".

Regarding claims 1, 15, and 16:

Hollander is directed to a method of simulating a system having a software component **(column 10 lines 24-28)** and a hardware component **(column 10 lines 24-28)**, said method comprising the steps of:

a. *modeling operation of said software component* using a software simulator **(column 8 lines 39-44; column 10 lines 51-58; column 6 lines 50-55);**

b. *modeling operation of said hardware component* using a hardware simulator **(column 8 lines 39-44; column 10 lines 51-58)**; wherein

c. said hardware simulator and said software simulator are *linked to model interaction* between said hardware component and said software component **(column 10 lines 34-49);**

d. generating with a test controller **(figure 1 controller 26; column 4 lines 66-67; column 5 line 1)** a *software stimulus* **(column 10 lines 59-61)** for said software component and a *hardware stimulus* **(column 2 lines 25-27; column 7 lines 12-14)** for said hardware component so as to permit *verification of correct interoperability* of said software component and hardware component **(column 10 lines 51-57),** wherein said modeled interaction between said components proceeds *independently of test controller* **(figure 1; column 5 lines 21-24; column 8 lines 24-32).**

Hollander fails to disclose a method where said software stimulus **(column 10 lines 59-61)** is passed to said software simulator **(figure 4 simulator 36)** by issuing a remote procedure call from said

test controller **(figure 1 controller 26; column 4 lines 66-67; column 5 line 1)** to said software simulator.

However **Platt teaches** an analogous method where a system call is made to execute a designated program, using the arguments supplied. When a program needs to be executed, the currently executing process makes a system call requesting that the program be executed. The scheduler makes an entry into the entry table and extracts the information needed to execute the program **(Platt: column 10 lines 16-22).**

It would have been obvious to one of ordinary skill in the art to combine the teachings of Hollander and Platt.

The motivation for doing so would have been to develop a method of communication that exhibits high performance **(Platt: column 7 lines 32-36).** While other communication protocols are cumbersome and time-consuming **(Platt: column 5 lines 12-15),** the method disclosed by Platt would overcome those deficiencies.

Regarding claim 2:

Platt teaches the use of a shared memory to store and retrieve information needed to execute the program. The user writes the information needed for the program execution to the main memory, and the scheduler extracts the information when it receives the request to execute the program **(Platt column 7 lines 1-26)**. The information needed to execute the program is analogous to the software stimuli in claim 2. The user is analogous to said test controller. The scheduler is analogous to said software simulator. The information extracted by the scheduler is analogous to the software stimuli.

Regarding claim 6:

Platt teaches that when a request is made for a program to be executed, the request is in the form of a process. The process consists of the executable program and all the information needed to run the program. The receiver of the request can then send back the appropriate response **(Platt: column 9 lines 34-36; column 10 lines 16-22).**

Regarding claim 7:

Hollander is directed to the method as claimed in claim 1 wherein said hardware component is a hardware peripheral within a data processing system **(column 11 lines 29-38).**

Regarding claim 8:

Hollander is directed to a method as claimed in claim 1 wherein said software component is a software driver for said hardware component **(column 11 lines 29-38; column 3 lines 10-13; column 6 lines 22-24)**.

Regarding claim 9:

Hollander is directed to a method as claimed in claim 1, further comprising monitoring modeled signals at an interface with said hardware component that are generated in response to simulation of said software component and said hardware component **(column 4 lines 52-56; column 8 lines 13-17)**.

Regarding claim 10:

Hollander is directed to a method as claimed in claim 9, wherein said modeled signals are monitored for compliance with rules defining permitted values for said modeled signals **(column 8 lines 13-17)**.

Regarding claim 11:

Hollander is directed to a method as claimed in claim 1, wherein said software simulator is monitored to determine coverage of a range of software stimuli that may be applied to said software simulator **(column 5 lines 32-33; column 8 lines 44-67; column 10 lines 51-55)**.

Regarding claim 12:

Hollander is directed to a method as claimed in claim 1, wherein said hardware simulator is monitored to determine coverage of a range of hardware stimuli that may be applied to said hardware simulator **(column 5 lines 32-33; column 8 lines 44-67; column 10 lines 51-55).**

Regarding claim 14

Hollander is directed to a method as claimed in claim 1, further compromising monitoring said hardware simulator to detect expected changes of state within said hardware component occurring in response to said software stimulus **(column 10 lines 51-68).**

2. **Claims 3-5** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Hollander (US Patent No 6,182,258)** and **Platt et al (US Patent No. 5,835,764)**, herein "Platt" as applied to claims 1-2 above and in further view of **Campbell (US Patent No. 6,408,009)**.

Regarding claim 3:

Hollander fails to disclose a method as claimed in claim 2 wherein said test controller sets a start flag within shared memory to indicate to said software simulator that said shared memory contains call data specifying a software stimulus be modeled.

Campbell teaches a method where a start flag in the program is used to inform the program later on that a condition has been met. In **figure 18**, steps 1000, 1002, 1004, and 1020 form that main part of the loop that relies on the start flag. The process starts at step 1000. In step 1002, the start flag is tested to determine whether or not it has been set high. If the start flag has not been set high, it loops back around to step 1002. If it has been set high, the rest of the program may execute, beginning with step 1004, and ultimately ending at step 1020 where the start flag is set low again to indicate that the program is finished executing.

It would have been obvious to combine the teachings of Hollander, Platt, and Campbell. Though Hollander and Platt describe a system to co-simulate a hardware and software component using appropriate communication protocols, both fail to teach what will trigger and control the flow of the program.

The motivation for doing so would be to incorporate an effective method for determining that the stimulus has been received and therefore the simulation can begin.

Regarding claim 4:

Campbell teaches a method as claimed in claim 3 wherein said software simulator polls said start flag to determine if there is a software stimulus to be modeled is contained in the above combination method of Hollander, Platt, and Campbell **(Campbell: figure 18 steps 1000, 1002)**

Regarding claim 5:

Campbell teaches a method as claimed in claim 3, wherein said software simulator resets said start flag to indicate to test controller that modeling of said software stimulus has been completed is contained in the above combination method of Hollander, Platt, and Campbell **(Campbell: figure 18 steps 1020)**.

3. **Claim 13** is rejected under 35 U.S.C. 103(a) as being unpatentable over **Hollander (US Patent No 6,182,258)** in view of **Platt et al (US Patent No. 5,835,764)**, herein "Platt" as applied to claim 1 above in further view of **Harmon (US Patent No. 6,810,373).**

Regarding claim 13:

Hollander fails to disclose a method as claimed in claim 1, wherein said software simulator is an instruction set simulator that serves to model execution of software program instruction by a data processing core.

Harmon teaches an analogous method wherein the co-verification environment contains an instruction set simulator (ISS) for representing the operation of the processor **(Harmon: column 3 lines 6-15)**. Therefore, the logic simulator models the hardware while the software is simultaneously modeled by the ISS.

At the time of the invention it would have been obvious to combine the teachings of Harmon with the above combination of Hollander and Platt.

The motivation for doing so would have been to increase the performance of the system. By using an ISS, the speed of verification is greatly increased **(Harmon: column 2 lines 66-67, column 3 lines 1-2, 46-57)**.

(10) Response to Argument

Response to Argument – Prior Art Rejections

Appellant's arguments pertaining to the 103 rejections are not persuasive. Appellant's arguments focusing on claims 1-16 are addressed in the order in which they are presented on the Appeal Brief starting on page 9.

The main thrust of Appellant's arguments center around Hollander's alleged failure to disclose modeling operation of a software component using a software simulator, having interaction between a software and hardware component that is independent from a test controller, and using a remote procedure call from the test controller to the software simulator. The Examiner emphasizes that the secondary reference (Platt) is relied upon only for the teaching of remote procedure calls. The Examiner maintains that the above features are taught by the combination of Hollander and Platt, as shown below.

(10.1) Appellant argues, on page 10, that Hollander does not teach modeling operation of a software component using a software simulator.

Examiner's Answer:

The portion of the Hollander reference cited in the Final Rejection (column 10 lines 51-58) discloses a co-verification module that allows the simultaneous verification of a software component and a hardware component. These two components make up the device-under-test (DUT) being tested. The prior art discloses that the DUT can be a complete system that comprises hardware with embedded logic (column 6 lines 50-55). Thus, the hardware component in the claim is analogous to the hardware portion of the DUT in the prior art and the software component in the language is analogous to the embedded logic in the DUT in the prior art. As shown in figure 1 of the prior art, the entire DUT is simulated in a simulation environment (figure 1 simulator 38 DUT 38). In order to simulate the entire DUT, the system must be able to simulate the individual modeled software and hardware components of the DUT. Thus, to perform co-verification, a hardware model is created, and the external software (which is not actual code as alleged by the Appellants, but is the modeled embedded logic of the DUT) is then run on the hardware model (i.e. both the modeled hardware and software components are simulated).

(10.2) Appellant argues, on page 12, that Hollander does not teach the claimed "generating" step wherein said modeled interaction between said software component and said hardware component proceeds independently of said test controller.

Examiner's Answer:

The Examiner asserts that the limitation 'interactions between said software component and said hardware component proceeds independently of said test controller' does not require that the interaction must *always* be independent. It merely requires that the interaction between the components be independent from the test controller *at some point*.

Column 5 lines 21-24 and column 8 lines 24-32 of the Hollander reference (as cited in the Final Rejection) states 'When using both dynamic generation and dynamic checking, the test generator module and the checker can constantly synchronize'. Thus, the two are required to synchronize (i.e. a dependent interaction) only when the system engages in both dynamic generation and dynamic checking; otherwise, the interaction between the components proceeds independently of said test controller.

The Examiner points to the last paragraph on page 12 of the Appeal Brief, which states that "this language [column 5 lines 21-24 of the Hollander reference] is broad enough to cover the Appellant's claimed subject matter".

The Appellant is further directed to column 7 lines 12-21 of the Hollander reference, which describes the test generator module. The test generator interacts with the variables of the HDL model, but there is no indication that it is dependent on the checker.

On page 13 of the Appeal Brief, the Appellant alleges that Figure 1 indicates that both the checker 30 and generator 26 are part of the test controller 22, and therefore it is impossible for the interaction to occur independently. The Appellant is directed to column 2 lines 25-29 of the Hollander reference, which defines a static testbench as a "program that drives pre-computed test vectors in the DUT simulator, and/or checks outputs after the simulation is completed" (emphasis added). Column 2

lines 35-38 of the reference further state, "...the internal state of the device at the point of error is not determined" (emphasis added). Thus, when performing static test generation and checking, the interaction between the components proceeds independently of the test controller because there is no interaction between the test controller and the components while the simulation is running.

(10.3) Appellant **argues, on page 14, that the Examiner fails to establish how or where Hollander teaches** Appellant's claimed modeling the response of the hardware component wherein software stimulus "is passed to said software simulator by issuing a remote procedure call from said test controller to said software simulator".

Examiner's Answer:

It is noted that the Examiner does not rely on Hollander for this teaching, and that pages 6-7 of the Final Rejection state that Hollander fails to disclose this limitation. The Examiner maintains that Platt teaches an analogous method where a system call is made to execute a designated program, using the arguments supplied. When a program needs to be executed, the currently executing process makes a system call requesting that the program be executed. The scheduler makes an entry into the entry table and extracts the information needed to execute the program (Platt: column 10 lines 16-22).

On page 15 of the Appeal Brief, the Appellant alleges that it would not be clear to one of ordinary skill in the art how to apply remote procedure calls a communication means in the Hollander reference, in view of the fact that the test controller and the external software exist in entirely separate domains with no apparent communication link available in order to implement the remote procedure call. The Examiner directs the Appellant to column 10 lines 34-64 of the Hollander reference, which discloses a communication link between the controller and software. In particular, the Examiner emphasizes the following portions:

- "A second protocol software layer is provided as a link, through the Unix socket, between the invention (interfaced with the DUT simulation) and the external software program" (lines 35-39)
- "One implication of this structure is that software verification is treaded in the context of the full verification effort. Thus, test generation, checking, debugging, coverage, and reporting all combine information from both the hardware and software sides" (column 10 lines 52-55)
- "...The invention's test generation facilities are available for the direct inputs to the DUT and the inputs to the external software" (lines 58-60)

Thus, Hollander discloses a communication link between the test controller and the external software, and Platt could be incorporated into the Hollander solution.

(10.4) Appellant argues, pages 18-19, the Examiner fails to establish any reason or motivation for combining references.

Examiner's Answer:

Appellant alleges that the motivation provided by the Examiner for the Hollander and Platt combination is inadequate, because the term 'high-performance' used in the cited portion of Platt (column 10 lines 16-22) is a term applied to almost every computer system in the world. Column 2 lines 39-42 of the Hollander reference states that static testbenches, while generally used, requires the expenditure of considerable time, especially during lost tests. Thus, Hollander expresses a need for a faster simulation, which would be provided by the use of remote of procedure calls, because this results in a high-performance system (Platt column 7 lines 32-36).

Appellant alleges that the Examiner's use of "effective method" is simply an indication that he can find no reason or motivation to combine the references. The Examiner asserts that the motivation to

combine the references centers on the need for an indication of when the stimulus is ready to be sampled. It is well known in the art that this can be effectively accomplished through the use of start flags and polling.

On page 21 of the Appeal Brief, the Appellant points out that the Examiner cites the Platt reference in the motivation statement to combine the Harmon reference with the Hollander and Platt references. The Examiner apologizes for this mistake, and asserts that the citation for the motivation statement should have read "(Harmon: column 2 lines 66-67; column 3 lines 1-2, 46-57)" instead of "(Platt: column 2 lines 66-67; column 3 lines 1-2, 46-57)". The cited portion of Harmon states, "...instruction-set simulators can run the software with much higher performance". The Examiner believes that the cited portion of the Harmon reference provides adequate motivation for the desired combination.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Shambhavi Patel, Patent Examiner

Conferees:

Kamini Shah, Supervisory Patent Examiner

Eddie Lee, Supervisory Patent Examiner